A high order numerical method for solving nonlinear fractional differential equation with non-uniform meshes ⋆

Lili Fan[1] and Yubin Yan[2][0000−0002−5686−5017]

[1] Department of Mathematics, Lvliang University, P. R. China 634936564@qq.com
[2] Department of Mathematics, University of Chester, Thornton Science Park, Pool Lane, Ince, CH2 4NU, UK, y.yan@chester.ac.uk

Abstract. We introduce a high-order numerical method for solving nonlinear fractional differential equation with non-uniform meshes. We first transform the fractional nonlinear differential equation into the equivalent Volterra integral equation. Then we approximate the integral by using the quadratic interpolation polynomials. On the first subinterval $[t_0, t_1]$, we approximate the integral with the quadratic interpolation polynomials defined on the nodes t_0, t_1, t_2 and in the other subinterval $[t_j, t_{j+1}], j = 1, 2, \ldots N-1$, we approximate the integral with the quadratic interpolation polynomials defined on the nodes t_{j-1}, t_j, t_{j+1}. A high-order numerical method is obtained. Then we apply this numerical method with the non-uniform meshes with the step size $\tau_j = t_{j+1} - t_j = (j+1)\mu$ where $\mu = \frac{2T}{N(N+1)}$. Numerical results show that this method with the non-uniform meshes has the higher convergence order than the standard numerical methods obtained by using the rectangle and the trapzoid rules with the same non-uniform meshes.

Keywords: Nonlinear fractional differential equation · Numerical method · Non-uniform meshes.

1 Introduction

Consider the following nonlinear fractional differential equation, with $\alpha > 0$,

$$ {}_0^C D_t^\alpha y(t) = f(t, y(t)),\ t > 0,\ y^{(k)}(0) = y_0^{(k)},\ k = 0, 1, \ldots, \lceil \alpha \rceil - 1, \qquad (1) $$

where ${}_0^C D_t^\alpha y(t)$ denotes the Caputo fractional derivative and $\lceil \alpha \rceil$ is the smallest integer $\geq \alpha$. Here $y_0^{(k)}$ are the initial values.

It is well-known that (1) is equivalent to, [4, Lemma 2.3]

$$ y(t) = \sum_{\nu=0}^{\lceil \alpha \rceil - 1} y_0^{(\nu)} \frac{t^\nu}{\nu!} + \frac{1}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} f(s, y(s))\, ds. \qquad (2) $$

⋆ Dr. Yubin Yan is the corresponding author.

For the existence and uniqueness of the solution of (1) and the application of the Newton iteration method for solving the nonlinear equation of the proposed numerical method, we demand that the function f is continuous on a suitable set $(0,T)\times(c,d)$ and $f(t,\cdot)\in C^2[c,d]$ for some $c,d\in\mathbb{R}^+$ and any fixed $t\in[0,T]$. Under these assumptions, Diethelm et al. [4, Theorems 2.1, 2.2] showed that (1) has a unique solution y on some interval $[0,T]$.

There are many works in the literature to consider the numerical methods for solving (1), see, e.g., [4], [2], [1], [8], [15], [16], [13]. Most numerical methods for solving (1) are designed and analyzed with the uniform meshes, see, e.g., [4], [5], [6], [8], [16]. Since the fractional differential equation is a nonlocal problem and the derivative of the solution of (1) has the singularity at $t=0$, it is not possible to obtain the high order numerical methods with uniform meshes. Therefore it is natural to use the non-uniform meshes to capture the singularity near $t=0$. Diethelm [3, Theorem 3.1] used the graded meshes to recover the optimal convergence order for the approximation of the Hadamard finite-part integral. Recently Stynes et al. [12], [11] applied the graded meshes to recover the convergence order of the finite difference method for solving time-fractional diffusion equation when the solution is not sufficiently smooth. Li et al. [7] considered the error estimates of the rectangle formula, trapezoid formula and the predictor-corrector scheme with non-uniform meshes for solving (1) under the assumption that the solution is sufficiently smooth. Other works for solving fractional differential equations with non-uniform meshes may be found in, for example, [7], [14], [17], [18].

Recently, Liu et al [9] designed a predictor-corrector numerical method for solving (1) with graded meshes and the detailed error estimates are provided. Liu et al [10] also introduced a numerical method with non-uniform meshes for solving (1) and the detailed error estimates are considered. This paper is the continuation of the works in [9], [10] and we will introduce a high order numerical method for solving (1) with non-uniform meshes. More precisely, we first approximate the integral in (2) with the piecewise quadratic interpolation polynomials with non-uniform meshes. We then use the Newton iteration for solving the nonlinear equation. Numerical examples show that this method has the high order convergence for solving nonlinear fractional differential equation with non-uniform meshes, where the solution of the fractional differential equation has the low regularity at $t=0$.

The novelties of this work are as follows:

1. A new way to approximate the integral in the Volterra integral equation (2) by using the piecewise quadratic polynomials is introduced.
2. A high order numerical method for solving nonlinear fractional differential equation (1) with non-uniform meshes is obtained which is particular useful when f is not sufficiently smooth with respect to time t. The convergence order of the proposed numerical method is higher than the available numerical methods in [7], [9], [10] for solving (1) with non-uniform meshes.

The paper is organized as follows. In section 2, we introduce a high-order numerical method for solving (1). In Section 3, we give a numerical example which shows that the numerical results are consistent with the theoretical results.

2 A high order numerical method with non-uniform meshes

For simplicity, we only consider the case with $\alpha \in (0,1)$ below. Similarly one may consider the general case with $\alpha > 1$. More precisely, we shall consider the numerical algorithm for solving, with $\alpha \in (0,1)$,

$$y(t) - y(0) = \frac{1}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} f(s, y(s))\, ds. \tag{3}$$

Let N be a positive integer. Let $0 = t_0 < t_1 < \cdots < t_n < t_{n+1} = T$, $n = 0, 1, 2, \ldots, N-1$ be the time partition of $[0, T]$. We want to find the approximate value y_{n+1} of $y(t_{n+1})$ at $t = t_{n+1}$. We shall consider the approximation of the following integral, with $n = 0, 1, 2, \ldots, N-1$,

$$I_{n+1} = \int_0^{t_{n+1}} (t_{n+1} - s)^{\alpha-1} f(s, y(s))\, ds = \sum_{j=0}^{n} \int_{t_j}^{t_{j+1}} (t_{n+1} - s)^{\alpha-1} f(s, y(s))\, ds.$$

It can be approximated by the following approach

$$I_{n+1} \approx \sum_{j=0}^{n} \int_{t_j}^{t_{j+1}} (t_{n+1} - s)^{\alpha-1} \tilde{f}_j(s, y(s))\, ds,$$

where $\tilde{f}_j(s, y(s)), j = 0, 1, 2, \ldots, n$ is the approximation of $f(s, y(s))$ on the interval $[t_j, t_{j+1}]$.

It will lead to different scheme by choosing different $\tilde{f}_j(s, y(s))$. In [7], Li et al. introduced the fractional rectangle, trapezoid, and predictor-corrector methods respectively. In this paper, we shall construct a high order numerical method for solving (2) by approximating the integral in (2) with the piecewise quadratic polynomials. Numerical examples in Section 3 show that the convergence order of the proposed numerical method is almost 3 for the sufficiently smooth function $f(t, y(t))$ and the suitable chosen non-uniform meshes as expected.

Let $P_2^{(j)}(s)$ denote the quadratic interpolation polynomial approximation of $f(s, y(s))$ on the interval $[t_j, t_{j+1}], j = 0, 1, 2, \ldots, n$, where $P_2^{(0)}(s)$ is the quadratic interpolation polynomial of $f(t, y(t))$ on the nodes t_0, t_1, t_2 and $P_2^{(j)}(s), j = 1, 2, \ldots, n$ are the quadratic interpolation polynomial of $f(t, y(t))$ on the nodes t_{j-1}, t_j, t_j. More precisely, we have

$$\begin{aligned} P_2^{(0)}(s) = {} & \frac{(s-t_1)(s-t_2)}{(t_0-t_1)(t_0-t_2)} f(t_0, y(t_0)) + \frac{(s-t_0)(s-t_2)}{(t_1-t_0)(t_1-t_2)} f(t_1, y(t_1)) \\ & + \frac{(s-t_0)(s-t_1)}{(t_2-t_0)(t_2-t_1)} f(t_2, y(t_2)), \end{aligned}$$

and, with $j = 1, 2, \ldots, n$,

$$\begin{aligned} P_2^{(j)}(s) = {} & \frac{(s-t_j)(s-t_{j+1})}{(t_{j-1}-t_j)(t_{j-1}-t_{j+1})} f(t_{j-1}, y(t_{j-1})) + \frac{(s-t_{j-1})(s-t_{j+1})}{(t_j-t_{j-1})(t_j-t_{j+1})} f(t_j, y(t_j)) \\ & + \frac{(s-t_{j-1})(s-t_j)}{(t_{j+1}-t_{j-1})(t_{j+1}-t_j)} f(t_{j+1}, y(t_{j+1})). \end{aligned}$$

We then get the following numerical approximate scheme for approximating (2),

$$y_{n+1} - y_0 = \sum_{j=0}^{n+1} \tilde{\tilde{w}}_{j,n+1} f(t_j, y_j), \tag{4}$$

where, after some simple but tedious calculations,

$$\frac{1}{\Gamma(\alpha+3)} \tilde{w}_{j,n+1} \tag{5}$$
$$= \begin{cases} \frac{2}{(t_0-t_1)(t_0-t_1)} C_0 & j=0, \\ \frac{2}{(t_1-t_0)(t_1-t_2)} C_1 + \frac{1}{(t_1-t_2)(t_1-t_3)} D_1 & j=1, \\ \frac{2}{(t_2-t_0)(t_2-t_1)} C_2 + \frac{1}{(t_2-t_1)(t_2-t_3)} D_2 + \frac{1}{(t_2-t_3)(t_2-t_4)} E_2 & j=2, \\ \frac{1}{(t_j-t_{j-2})(t_j-t_{j-1})} C_j + \frac{1}{(t_j-t_{j-1})(t_j-t_{j+1})} D_j + \frac{1}{(t_j-t_{j+1})(t_j-t_{j+2})} E_j & j=3,4,\ldots,n-1, \\ \frac{1}{(t_n-t_{n-2})(t_n-t_{n-1})} D_n + \frac{1}{(t_n-t_{n-1})(t_n-t_{n+1})} E_n & j=n, \\ \frac{1}{(t_{n+1}-t_{n-1})(t_{n+1}-t_n)} E_{n+1} & j=n+1. \end{cases}$$

Here

$$\begin{aligned} C_0 =& \alpha(\alpha+1)\big[(t_{n+1}-t_0)^{\alpha+2} - (t_{n+1}-t_1)^{\alpha+2}\big] \\ & - \alpha(\alpha+2)(2t_{n+1}-t_1-t_2)\big[(t_{n+1}-t_0)^{\alpha+1} - (t_{n+1}-t_1)^{\alpha+1}\big] \\ & + (\alpha+1)(\alpha+2)(t_{n+1}-t_1)(t_{n+1}-t_2)\big[(t_{n+1}-t_0)^{\alpha} - (t_{n+1}-t_1)^{\alpha}\big], \end{aligned}$$

$$\begin{aligned} C_1 =& \alpha(\alpha+1)\big[(t_{n+1}-t_0)^{\alpha+2} - (t_{n+1}-t_1)^{\alpha+2}\big] \\ & - \alpha(\alpha+2)(2t_{n+1}-t_0-t_2)\big[(t_{n+1}-t_0)^{\alpha+1} - (t_{n+1}-t_1)^{\alpha+1}\big] \\ & + (\alpha+1)(\alpha+2)(t_{n+1}-t_0)(t_{n+1}-t_2)\big[(t_{n+1}-t_0)^{\alpha} - (t_{n+1}-t_1)^{\alpha}\big] \end{aligned}$$

$$\begin{aligned} D_1 =& \alpha(\alpha+1)\big[(t_{n+1}-t_2)^{\alpha+2} - (t_{n+1}-t_3)^{\alpha+2}\big] \\ & - \alpha(\alpha+2)(2t_{n+1}-t_2-t_3)\big[(t_{n+1}-t_2)^{\alpha+1} - (t_{n+1}-t_3)^{\alpha+1}\big] \\ & + (\alpha+1)(\alpha+2)(t_{n+1}-t_2)(t_{n+1}-t_3)\big[(t_{n+1}-t_2)^{\alpha} - (t_{n+1}-t_3)^{\alpha}\big], \end{aligned}$$

$$\begin{aligned} C_2 =& \alpha(\alpha+1)\big[(t_{n+1}-t_0)^{\alpha+2} - (t_{n+1}-t_1)^{\alpha+2}\big] \\ & - \alpha(\alpha+2)(2t_{n+1}-t_0-t_1)\big[(t_{n+1}-t_0)^{\alpha+1} - (t_{n+1}-t_1)^{\alpha+1}\big] \\ & + (\alpha+1)(\alpha+2)(t_{n+1}-t_0)(t_{n+1}-t_1)\big[(t_{n+1}-t_0)^{\alpha} - (t_{n+1}-t_1)^{\alpha}\big], \end{aligned}$$

$$\begin{aligned} D_2 =& \alpha(\alpha+1)\big[(t_{n+1}-t_2)^{\alpha+2} - (t_{n+1}-t_3)^{\alpha+2}\big] \\ & - \alpha(\alpha+2)(2t_{n+1}-t_1-t_3)\big[(t_{n+1}-t_2)^{\alpha+1} - (t_{n+1}-t_3)^{\alpha+1}\big] \\ & + (\alpha+1)(\alpha+2)(t_{n+1}-t_1)(t_{n+1}-t_3)\big[(t_{n+1}-t_2)^{\alpha} - (t_{n+1}-t_3)^{\alpha}\big], \end{aligned}$$

$$\begin{aligned}E_2 =&\alpha(\alpha+1)\big[(t_{n+1}-t_3)^{\alpha+2}-(t_{n+1}-t_4)^{\alpha+2}\big]\\&-\alpha(\alpha+2)(2t_{n+1}-t_3-t_4)\big[(t_{n+1}-t_3)^{\alpha+1}-(t_{n+1}-t_4)^{\alpha+1}\big]\\&+(\alpha+1)(\alpha+2)(t_{n+1}-t_3)(t_{n+1}-t_4)\big[(t_{n+1}-t_3)^{\alpha}(t_{n+1}-t_4)^{\alpha}\big],\end{aligned}$$

$$\begin{aligned}C_j =&\alpha(\alpha+1)\big[(t_{n+1}-t_j)^{\alpha+2}-(t_{n+1}-t_{j+1})^{\alpha+2}\big]\\&-\alpha(\alpha+2)(2t_{n+1}-t_{j-2}-t_{j-1})\big[(t_{n+1}-t_j)^{\alpha+1}-(t_{n+1}-t_{j+1})^{\alpha+1}\big]\\&+(\alpha+1)(\alpha+2)(t_{n+1}-t_{j-2})(t_{n+1}-t_{j-1})\big[(t_{n+1}-t_j)^{\alpha}-(t_{n+1}-t_{j+1})^{\alpha}\big],\end{aligned}$$

$$\begin{aligned}D_j =&\alpha(\alpha+1)\big[(t_{n+1}-t_j)^{\alpha+2}-(t_{n+1}-t_{j+1})^{\alpha+2}\big]\\&-\alpha(\alpha+2)(2t_{n+1}-t_{j-1}-t_{j+1})\big[(t_{n+1}-t_j)^{\alpha+1}-(t_{n+1}-t_{j+1})^{\alpha+1}\big]\\&+(\alpha+1)(\alpha+2)(t_{n+1}-t_{j-1})(t_{n+1}-t_{j+1})\big[(t_{n+1}-t_j)^{\alpha}-(t_{n+1}-t_{j+1})^{\alpha}\big],\end{aligned}$$

$$\begin{aligned}E_j =&\alpha(\alpha+1)\big[(t_{n+1}-t_j)^{\alpha+2}-(t_{n+1}-t_{j+1})^{\alpha+2}\big]\\&-\alpha(\alpha+2)(2t_{n+1}-t_{j+1}-t_{j+2})\big[(t_{n+1}-t_j)^{\alpha+1}-(t_{n+1}-t_{j+1})^{\alpha+1}\big]\\&+(\alpha+1)(\alpha+2)(t_{n+1}-t_{j+1})(t_{n+1}-t_{j+2})\big[(t_{n+1}-t_j)^{\alpha}-(t_{n+1}-t_{j+1})^{\alpha}\big],\end{aligned}$$

$$\begin{aligned}D_n =&\alpha(\alpha+1)\big[(t_{n+1}-t_{n-1})^{\alpha+2}-(t_{n+1}-t_n)^{\alpha+2}\big]\\&-\alpha(\alpha+2)(2t_{n+1}-t_{n-2}-t_{n-1})\big[(t_{n+1}-t_{n-1})^{\alpha+1}-(t_{n+1}-t_n)^{\alpha+1}\big]\\&+(\alpha+1)(\alpha+2)(t_{n+1}-t_{n-2})(t_{n+1}-t_{n-1})\big[(t_{n+1}-t_{n-1})^{\alpha}-(t_{n+1}-t_n)^{\alpha}\big],\end{aligned}$$

$$\begin{aligned}E_n =&\alpha(\alpha+1)(t_{n+1}-t_n)^{\alpha+2}-\alpha(\alpha+2)(t_{n+1}-t_{n-1})(t_{n+1}-t_n)^{\alpha+1}\\&+(\alpha+1)(\alpha+2)(t_{n+1}-t_{n-1})(t_{n+1}-t_n)^{\alpha},\end{aligned}$$

$$\begin{aligned}E_{n+1} =&\alpha(\alpha+1)(t_{n+1}-t_n)^{\alpha+2}-\alpha(\alpha+2)(2t_{n+1}-t_{n-1}-t_n)(t_{n+1}-t_n)^{\alpha+1}\\&+(\alpha+1)(\alpha+2)(t_{n+1}-t_{n-1})(t_{n+1}-t_n)(t_{n+1}-t_n)^{\alpha}.\end{aligned}$$

Now we need to solve y_{n+1} in (4) with the weights defined in (5). Let us here only consider how to calculate y_1. The calculation of $y_l, l \geq 2$ is similar. Note that, by (4),

$$y_1 = y_0 + \tilde{\tilde{w}}_{0,1} f(t_0, y_0) + \tilde{\tilde{w}}_{1,1} f(t_1, y_1).$$

Denote

$$g(y_1) = y_1 - [y_0 + \tilde{\tilde{w}}_{0,1} f(t_0, y_0) + \tilde{\tilde{w}}_{1,1} f(t_1, y_1)].$$

We then need to solve $g(y_1) = 0$ which is a nonlinear equation with respect to the variable y_1. Let $z_0 = y_0$ be the initial guess, then y_1 can be approximated by $z_M \approx y_1$ which is obtained by the following Newton iteration formula

$$z_{l+1} = z_l - \frac{g(z_l)}{g'(z_l)},\ l = 0, 1, 2, \ldots, M.$$

Here g' denotes the derivative of g and the positive integer $M \in \mathbb{N}$ can be determined by using the error control quantity $|z_M - z_{M-1}| < 10^{-10}$. The assumption for f in our paper guarantees that the sequence $z_l, l = 0, 1, 2, \ldots$ is convergent.

Remark 1. The work in this paper is the extension of the work in [7] where the authors introduced the fractional rectangle, trapezoid and predictor-corrector methods with non-uniform meshes for solving (1). The stability and error estimates are discussed in [7]. One may use the similar approach to discuss the stability and error estimates of the proposed numerical method (4) in this work.

3 Numerical Results

We will now look at some numerical results for the numerical method defined in (4) with non-uniform mesh with the time step size

$$\tau_j = t_{j+1} - t_j = (j+1)\mu,\ j = 0, 1, 2, \ldots, N-1, \tag{6}$$

where $\mu = \frac{2T}{N(N+1)}$.

Remark 2. Following the analysis in [7, Section 4], if $f(t, y(t))$ is sufficiently smooth, the convergence orders of the proposed numerical methods in [7] for both uniform and non-uniform meshes are highly possible the same. But for non-smooth function $f(t, y(t))$, non-uniform meshes are much suitable than uniform meshes. For the non-smooth function $f(t, y(t))$, Li et al. [7] proved the error estimates for the fractional rectangle, trapezoid and predictor-corrector methods with the concrete non-uniform meshes (6). The mesh (6) is not the unique non-uniform meshes in literature. In general one may consider the graded meshes with $t_j = T(j/N)^r, r > 0$, see, e.g., [9], [10]. In fact, after a simple calculation, one may see that the mesh (6) is equivalent to the graded mesh with $r = 2$. In some cases, one may get better convergence order when choosing the graded mesh with $r > 2$ for some non-smooth function $f(t, y(t))$.

In this section, we shall consider two examples. In the first example, we consider the case where the solution of (1) is very smooth and in the second example we consider the case where the solution is less regular.

Example 1. Consider, with $\alpha \in (0, 1)$ and $\beta > 0$,

$${}_0^C D_t^\alpha y(t) = \frac{\Gamma(1+\beta)}{\Gamma(1+\beta-\alpha)} t^{\beta-\alpha} + t^{2\beta} - y^2, \tag{7}$$

where $f(t, y) = \frac{\Gamma(1+\beta)}{\Gamma(1+\beta-\alpha)} t^{\beta-\alpha} + t^{2\beta} - y^2$ and the exact solution is $y(t) = t^\beta$. We choose $\beta = 2$ and the exact solution is now very smooth $y(t) = t^2$.

In Tables 1, we list the experimentally determined convergence orders for the quadratic method (4) with respect to the different $\alpha = 0.4, 0.6, 0.8$. We observe that the quadratic method with the non-uniform meshes has the convergence order almost 3 as we expected.

Meshes	N	$\alpha = 0.4$	EOC	$\alpha = 0.6$	EOC	$\alpha = 0.8$	EOC
Uniform	40	2.61E-06		6.85EE-07		3.70E-06	
	80	6.56E-07	1.99	1.75E-07	1.97	9.26E-07	1.99
	160	1.64E-07	1.99	4.39E-08	1.99	2.32E-07	1.99
	320	4.10E-08	1.99	1.10E-08	1.99	5.79E-08	2.00
	640	1.03E-08	2.00	2.74E-09	2.00	1.45E-08	2.00
Non-Uniform	40	1.58E-06		2.09E-06		1.80E-06	
	80	2.01E-07	2.98	2.59E-07	3.00	2.21E-07	3.02
	160	2.54E-08	2.98	3.23E-08	3.00	2.73E-08	3.01
	320	3.21E-09	2.99	4.03E-09	3.00	3.39E-09	3.00
	640	4.04E-10	2.99	5.04E-10	3.00	4.24E-10	3.00

Table 1. Errors at T=1 by using quadrature method (4) in Example 1

Example 2. We consider the same equation as in Example 1 with $\beta = 0.9$. In this case the exact solution is $y(t) = t^{0.9}$ which is not so regular. In Table 2, we observe that the convergence orders are much lower than the smooth case in Example 1 for both uniform and non-uniform meshes. This is because $f(t, y(t))$ behaves as $t^{\beta-\alpha}, \beta = 0.9$ in Example 2 which is less smoother than $t^{2-\alpha}$ in Example 1. The convergence order depends on the smoothness of the regularity of $f(t, y(t))$, see [7], [9], [10].

Meshes	N	$\alpha = 0.4$	EOC	$\alpha = 0.6$	EOC	$\alpha = 0.8$	EOC
Uniform	40	4.42E-04		1.91E-03		5.76E-03	
	80	2.44E-04	0.86	1.04E-03	0.88	3.11E-03	0.89
	160	1.32E-04	0.88	5.59E-04	0.89	1.67E-03	0.89
	320	7.13E-04	0.89	2.99E-04	0.90	8.94E-04	0.90
	640	3.83E-05	0.90	1.61E-04	0.90	4.79E-04	0.90
Non-Uniform	40	1.03E-04		2.43E-04		4.58E-04	
	80	2.97E-05	1.79	6.98E-05	1.79	1.31E-04	1.79
	160	8.54E-06	1.80	2.01E-05	1.79	3.78E-05	1.80
	320	2.45E-06	1.80	5.76E-06	1.80	1.08E-05	1.80
	640	7.04E-07	1.80	1.65E-06	1.80	3.11E-06	1.80

Table 2. Errors at T=1 by using quadrature method (4) in Example 2

References

1. Cao, J., Xu, C.: A high order schema for the numerical solution of the fractional ordinary differential equations. J. Comp. Phys. **238**, 154–168 (2013)
2. Deng, W.H.: Short memory principle and a predict-corrector approach for fractional differential equations. J. Comput. Appl. Math. **206**, 1768–1777 (2007)
3. Diethelm, K.: Generalized compound quadrature formulae for finite-part integral. IMA J. Numer. Anal. **17**, 479–493 (1997)

4. Diethelm, K., Ford, N.J.: Analysis of fractional differential equations. J. Math. Anal. Appl. **265**, 229–248 (2002)
5. Diethelm, K., Ford, N.J., Freed, A.D.: Detailed error analysis for a fractional Adams method. Numerical Algorithms **36**, 31–52 (2004)
6. Diethelm, K., Ford, N.J., Freed, A.D.: A predictor-corrector approach for the numerical solution of fractional differential equations. Nonlinear Dynamics **29**, 3–22 (2002)
7. Li, C., Yi, Q., Chen,A.: Finite difference methods with non-uniform meshes for nonlinear fractional differential equations. Journal of Computational Physics **316**, 614–631 (2016)
8. Li.C., Zeng, F.: The finite difference methods for fractional ordinary differential equations. Numer. Funct. Anal. Optim. **34**, 149–179 (2013)
9. Liu, Y., Roberts, J., Yan, Y.: A note on finite difference methods for nonlinear fractional differential equations with non-uniform meshes, International Journal of Computer mathematics **95**, 1151–1169 (2018)
10. Liu, Y., Roberts, J., Yan, Y.: Detailed error analysis for a fractional Adams method with graded meshes. Numer. Algor. (2017), https://doi.org/10.1007/s11075-017-0419-5
11. Stynes, M.: Too much regularity may force too much uniqueness. Fractional Calculus and Applied Analysis **19**, 1554–1562 (2016)
12. Stynes, M., O'riordan, E., Gracia, J.L.: Error analysis of a finite difference method on graded meshes for a time-fractional diffusion equation. SIAM J. Numer.Anal. **55**, 1057–1079 (2017)
13. Pal, K., Liu, F., Yan,Y.: Numerical solutions for fractional differential equations by extrapolation. Lecture Notes in Computer Science, Springer series, 9045, 299–306 (2015)
14. Quintana-Murillo, J., Yuste, S.B.: A finite difference method with non-uniform timesteps for fractional diffusion and diffusion-wave equations. The European Physical Journal Special Topics, **222**, 1987–1998 (2013)
15. Yan, Y., Pal, K., Ford, N.J.: Higher order numerical methods for solving fractional differential equations. BIT Numer. Math., **54**, 555–584 (2014)
16. Zhao, L., and Deng, W.H.: Jacobi-predictor-corrector approach for the fractional ordinary differential equations. Advances in Computational Mathematics **40**, 137–165 (2014)
17. Yuste, S.B., Quintana-Murillo, J.: Fast, accurate and robust adaptive finite difference methods for fractional diffusion equations. Numer. Algor., **71**, 207–228 (2016)
18. Zhang, Y., Sun, Z., Liao, H.: Finite difference methods for the time fractional diffusion equation on non-uniform meshes. Journal of Computational Physics **265**, 195–210 (2014)